

**Continuous Speech Recognition in a Language Tutor – Using
Learning Principles to Alleviate Underlying Problems**

Jonathan D. Kaplan and V. Melissa Holland
U.S. Army Research Institute

19981204 052

Advanced Training Methods Research Unit
Franklin L. Moses, Chief

December 1998

**Reproduced From
Best Available Copy**



**U.S. Army Research Institute
for the Behavioral and Social Sciences**

Approved for public release; distribution is unlimited.

**U.S. Army Research Institute
for the Behavioral and Social Sciences**

A Directorate of the U.S. Total Army Personnel Command

EDGAR M. JOHNSON
Director

NOTICES

DISTRIBUTION: This Research Note has been cleared for release to the Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or the National Technical Information Service (NTIS).

FINAL DISPOSITION: This Research Note may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The views, opinions, and findings in this Research Note are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy) 12-01-98	2. REPORT TYPE Final	3. DATES COVERED (from... to) 04-09-92 to 12-01-98		
4. TITLE AND SUBTITLE Continuous Speech Recognition in a Language Tutor—Using Learning Principles to Alleviate Underlying Problems			5a. CONTRACT OR GRANT NUMBER	
			5b. PROGRAM ELEMENT NUMBER 0602785A	
6. AUTHOR(S) Jonathan D. Kaplan, V. Melissa Holland			5c. PROJECT NUMBER A790	
			5d. TASK NUMBER 2233	
			5e. WORK UNIT NUMBER CO4	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences ATTN: PERI-II 5001 Eisenhower Avenue Alexandria, VA 22333-5600			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue Alexandria, VA 22333-5600			10. MONITOR ACRONYM ARI	
			11. MONITOR REPORT NUMBER Research Note 99-05	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES				
<p>14. ABSTRACT (Maximum 200 words):</p> <p>This paper describes the instructional features of the Military Language Tutor (MILT), how they were shaped by principles of learning and memory drawn from work in experimental psychology, and how these approaches are being used to deal with the problems of continuous speech recognition in a tutor.</p>				
15. SUBJECT TERMS authoring, foreign language, learning theory, NLP, speech recognition, intelligent tutor, tutor, Military Language Tutor, MILT				
16. REPORT Unclassified			17. ABSTRACT Unclassified	18. THIS PAGE Unclassified
19. LIMITATION OF ABSTRACT Unlimited			20. NUMBER OF PAGES 23	21. RESPONSIBLE PERSON (Name and Telephone Number) Jonathan D. Kaplan (703) 617-8828

Continuous Speech Recognition in a Language Tutor— Using Learning Principles to Alleviate Underlying Problems

Jonathan D. Kaplan and V. Melissa Holland

U.S. Army Research Institute

Introduction

The Army Research Institute (ARI) has begun a project known as the Military Language Tutor (MILT) to develop speech driven, graphics language tutor that gives job-relevant communicative practice. Our paper will describe the instructional features of MILT-CSR (continuous speech recognition) and explain how certain of these features are shaped by principles of learning and memory drawn from work in experimental psychology, and how these approaches are being used to deal with the problems of continuous speech recognition in a tutor. As psychologists in human factors and instruction, we approach the problem of designing a speech driven language tutor from a somewhat different angle than do others in this volume. We and our colleagues have had considerable experience developing training systems in areas other than foreign language, in which we have applied general principles from research on motivation, cognition, skills acquisition and retention, and human factors. The application of these principles has resulted in demonstrably effective programs, models, and devices now used in the military as well as, in various transformations, in industry and schools (Berkowitz & Simutis, 1983; Farr & Ward, 1988; Hagman, Hayes, & Bierwirth, 1986; Hagman & Rose, 1983; Kaplan, 1988; Laughery, Dahl, Kaplan, Archer, & Fontenelle, 1988; Oxford, Harman, & Holland, 1987; Psotka, Massey, & Mutter, 1988; Wisher, Sabol, & Kern, forthcoming ; Wisher, Holland, & Chatelier, 1987; Yates & Macpherson, 1985).

Application of these principles to a language tutor assumes that the language in use shares features with human skills like adding numbers or driving a car, rather than being an exclusive realm of knowledge with unique organizational and acquisitional principles (for a recent articulation of the language-as-skill point of view, see Lawton & Andreson, XXX). Theoretical claims about the nature of language are widely disputed, and we do not wish to engage in those disputes here, but rather to draw on fruitful analogies that yield sensible ideas for the design of our tutor.¹

Some of the principles that underlie MILT-CSR2 derive from behavioristically oriented learning theory. One of the early foundations of experimental psychology, this theory attempts to represent the dynamics of learning, some of which were originally described by Skinner (1938) and others through animal experiments. While some of

¹ Even the most extreme ontological position in regard to language—that its structure is innate and universal and governed by laws of acquisition not shared by cognitive and physical domains (Chomsky, 1981)—does not preclude appeal to general cognitive principles to explain the learning of language-specific vocabulary, constructions, and rules of use.

these findings and principles are of dubious generalizability, others have become accepted bases for the design of computer-based training and programmed instruction (Anderson, Kulhavy, & Andre, 1971; Gagne & Briggs, 1979). Learning theory has pointed to practice and reinforcement as major dynamics, and these continue as significant constructs in cognitive formulations of learning (Anderson & Schooler, 1991; Newell, 1990; Schmidt & Bjork, 1992), with reinforcement replaced by concepts of feedback and knowledge of results (Schmidt, 1990).

The MILT-CSR system draws on ARI's earlier BRIDGE and MILT projects (Kreyer & Criswell, 1995; Sams, 1995), but neither of these projects made use of speech recognition. MILT-CSR is currently under development. A full text input version exists and a discrete speech input version will be completed in 1996. A continuous speech recognition version will begin development in 1997. The current MILT-CSR text input tutor overlies a natural language processing (NLP) engine that uses parsing mechanisms described by Weinberg, Garman, Martin, and Merlo (1995) and that incorporates semantic and dialogue analysis components discussed by Dorr, Helder, Blanksteen, and Migdaloff (1995). The question of using this parsing machinery with speech recognition will be discussed later in this paper.

Speech Recognition and MILT-CSR

Background

Language usage consists of some combination of language production and understanding, according to some combination of text and speech modalities. In general, language production is considered to be more difficult to learn than understanding within a given modality. Outside the academic community, the speech mode of communication is considered more important than the text mode. So, it is reasonable to suggest that the most difficult and desired language skill is speech production. If a tutor asks a student to read aloud the correct written statement which is one of a set of alternatives, the skill involved is still language understanding, not production. It is true that pronunciation skills will be exercised, but pronunciation of read material is not production.

There is a major benefit to developing a "select-and-read-aloud" tutor of this kind using speech recognition. Since the written alternatives were created by an author who deliberately introduced some kinds of errors into the incorrect alternatives, the tutor system knows what the real errors are. Speech recognition doesn't have to identify these errors itself. All speech recognition has to do is identify which of the written alternatives are being spoken. Even if speech recognition alters the specific words, some form of gisting or word spotting will probably be able to match the utterance to the alternative with its known errors. Such a tutor would be quite useful for providing practice and testing recognition and pronunciation, but it would not be able to teach language production except in the most indirect way. Language

production is spontaneous speech or writing. It may be in response to a question or cue, but it is not copying or word for word transcription from one medium to another.

The General Problem. It is difficult for people to correct mistakes if they do not know they are making them. There are several major categories of language usage (as opposed to factual) mistakes that people make—grammatical/syntactic, semantic/usage, vocabulary, and pronunciation mistakes. To the extent that continuous speech recognition (CSR) has been considered for use in language tutors, its major role has been in improving pronunciation by using native speaker language models and comparing student pronunciation to that of the native speakers, and as a device to evaluate reading. These approaches play to the current strengths of CSR. However, the recognition of grammatical and semantic errors requires a process that plays to CSR's current weakness. To recognize student's grammatical and semantic mistakes, CSR would have to recognize exactly what the student really said, word for word. It could not describe the sense of what the student meant to say based on a grammatically correct language model without losing the grammatical errors that the student made.

To recognize exactly what a student says, a CSR would need a language model created from realistic student data, some mechanism for altering a conventional error-free language model according to expected errors, or it would have to rely significantly more than usual on an acoustic model.

The Problems of Using a Student Language Model. Since students are expected to improve as they use a tutor and thus reduce errors, the student model would somehow have to take account of this. There are two major classes of problems connected with solving the CSR problem with a student language model—size of required random access memory (RAM), and cost of data collection. When you consider that a current normal speech CSR requires approximately 90MB of RAM to make accessible 20,000 words, the magnitude of the RAM problem becomes evident. One approach to the RAM problem might be an intelligent student language model. That is, the model would alter according to cues it received from the student's speech input. At the moment, no one knows exactly how to construct such an intelligent language model. A simpler, more brute force approach would be for the language model simply to include all significant variations of errors from beginner to advanced. If you built a large scale language model on the order of 20,000 words, and you included all major classes of errors, it is likely the RAM requirements would be larger than could be practically met. You could build a much more constrained language model on the order of 1-2,000 words and then the resulting explosion in RAM required for error effects might still be practical. Collecting large scale speech data for students of varying language ability is possible, but quite expensive since you would have to collect it from speakers at all levels of interest from those who produce the largest numbers of errors to those who produce none. Depending on the range of student ability of interest, such an undertaking would be the equivalent of collecting multiple native speaker language data bases. Once again, the size of this undertaking could be

reduced by highly constraining the language model, but the utility of the resulting speech model(s) would also be reduced.

The Problems of Perturbing a Native Speaker Language Model. The object of this process is to make a native speaker language model, which is not based on error containing student speech data, behave as if these error data were present. On the one hand, this can be done by manually altering the native speaker language model according to an analysis of student errors. Such an analysis could result from questioning instructors or taking data from students. On the other hand, it can be done by introducing a natural language processing engine (NLP) that has been developed to identify syntactic and/or semantic errors into the CSR process. Such an NLP could be introduced in the front end of the CSR process to help identify word input and the back end to identify errors. When the speech model predicts what a given word is, it would send this prediction to the NLP which would then confirm that prediction or make an alternative prediction which it would feed back to the model for confirmation or rejection. Given that an error identifying NLP were used, its rules could be used to identify speech input errors for each sentence. The good news is that this approach should use substantially less RAM and in the long run cost fewer dollars than developing a brute force mega model containing all likely errors in combinations according to student language level. The bad news is that it has never been tried, and no one is sure how to do it.

The Problems of Using an Acoustic Model. The best approach to using CSR to identify the actual syntactic and semantic errors made by speakers should be by using only an acoustic model since it would identify the exact words spoken based upon their acoustic signature. Unfortunately, this pure approach results in an unacceptably high error rate. The reason for using the language model with the acoustic model is to use the language model's statistical predictions to lower the error rate to something more nearly acceptable. In theory, one might be able to raise the sampling rate of the acoustic model to such an extent that its accuracy became acceptable. Unfortunately, with current computers if this were possible it would require so much computer time to do the analysis as to make the process totally impractical. Another approach would be to use an syntactic/semantic error prediction method (see [The Problems of Perturbing a Native Speaker Language Model, above](#)) to turn up the acoustic model's sampling rate only when such an error was predicted and at the same time to turn off the language model. Once again, the potential solutions to the problems of increasing the use of the acoustic model have not been tried, and no one is quite sure how to implement them.

General Psychological Principles upon which MILT-CSR Is Based

Practice and Intrinsic Motivation

The initial target users for MILT-CSR are Special Forces soldiers who use foreign language in their training and quasi-diplomatic functions. They have already received language training, and they can be assumed to have reached a Level 2 in target language proficiency (on the 5-point Interagency Language Roundtable scale). As

noted by Sams (1995), these soldiers are more likely to use a language tutor for optional self-study than as part of a formal program of instruction. For this reason, we wanted the tutor to be interesting enough that students would want to use it on their own time, and, moreover, would want to explore the tutoring environment beyond the basic exercises. The learning principle here is straightforward: The more learners use language—to the point of overlearning words and constructions—the better they will retain it. According to learning theory, practice improves performance (Anderson & Schooler, 1991; Schmidt & Bjork, 1992). Learning has been shown to follow some form of the classical learning curve, a power function schematized in Fig 1.

The learning curve indicates that after some amount of practice, performance will be as good as it will get, and there wouldn't be much point in practicing more. But this doesn't take into account a phenomenon called *overlearning*. It turns out that when well-trained people are placed under significant stress, they experience surprising breakdowns in performance. And if people do not practice a skill regularly, they tend to forget it. However, if they have practiced that skill many times after they already

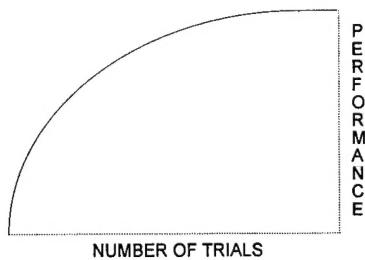


Figure 1. Typical learning curve.

appear to be performing well, then the stress-related breakdowns and the losses over time are reduced—thus overlearning (Driskell, Willis, & Copper, 1992; Kreuger, 1929; Schendel & Hagman, 1982). Since the purpose of language training in the military is to enable realistic performance in situations that are sometimes stressful, and to support retention of language skills over periods of nonusage, it is desirable to go past the point where the learning curve asymptotes. Therefore, the key to improving performance is to design a tutor that so interacts with students as to produce successful practice trials beyond the level at which students manifest mastery of the material.

How could we design a tutor that would motivate students to practice on their own time and to continue beyond basic exercises? We first reasoned that students would be encouraged toward self-study by a system that is intrinsically rewarding.

Intrinsic motivation. It is well established in behavioral research that rewards improve learning (Deese & Carpenter, 1951; Miller, 1963). Typically, people think of reward as some action that occurs upon the successful completion of some behavior. The rat gets to the end of the maze and is given cheese. The language student practices conjugating in the future perfect tense, takes a test, and is given an *A*. This is extrinsic

reward. The behavior itself is not rewarding. It is the cheese, and the *A*, delivered from an external source that reinforces the desired behavior, maze running and conjugation practicing.

Extrinsic reward does not explain the behavior of people endlessly playing computer games that they do not win, or artists who paint without hope of selling, or people who do crossword puzzles without being able to finish them. In these examples, people are motivated intrinsically, by the pleasure of the behavior itself. The facilitating effects of intrinsic motivation on learning have been widely demonstrated (Berlyne, 1968; McClelland, 1961; Malone, 1981). The behavior that people find intrinsically motivating appears to be quite diverse. However, much of this apparent diversity has underlying commonality. For example, common elements are exercising control over external objects and engaging in problem solving.

In the case of a language tutor, we interpreted intrinsic reward to mean constructing exercises that offer students the opportunity to practice in the context of doing something that is important or interesting to them and at which they can succeed. We assume that they will be rewarded by some version of controlling their environment and solving problems, but the specific form that these two dimensions will take differs according to who the learners are.

Application of intrinsic motivation in the ARI language tutor. We know that the military linguists who make up our user population are more intelligent than the average person, motivated to learn and retain foreign languages, job oriented, but likely to become bored. Taking these characteristics into account, we developed a design for MILT-CSR that can support intrinsic reward through several means.

The types of exercises being built into MILT-CSR enable realistic communicative interactions in which students solve an interesting problem or execute a simulated job task. Such interactions can be interpreted as applying a communicative approach to language teaching, as discussed by Douglas (1995), whose LingWorlds tutor provides a precedent for using language to solve nonlanguage problems. In MILT-CSR the primary exercise type designed to motivate students to want to use and improve their language is an animated graphics microworld.

The Reason that Error Identification is Important. Work in instructional psychology (e.g., Levine, 1975) suggests that learners' performance can be improved with appropriate feedback that points out errors. This finding has received recent support in the area of second language learning (e.g., Lightbown & Spada, 1990). The instructional psychology literature further suggests that informative feedback—telling learners the nature of their error rather than simply that there is an error—works better for most learners (Kulhavy, 1977; Kulik & Kulik, 1988). At the same time, this literature suggests that high-ability learners who are also field-independent (a cognitive style characteristic grounded in research by Witkins, et al., 1977) may do better if they have to figure out their error—that is, if they receive simple "uninformative" feedback. Thus, knowledge of results appears to be a useful parameter to manipulate in a tutoring system.

It is the pattern of these errors that may be the best metric of learners' progress, and the resulting feedback that we assume to be critical to mastering language form and content. We do not in this tutor commit to either a form-focused or a communicative approach; instead, either approach, or compromises between them, can be instantiated through the authoring interface. As pointed out for the dialogue and microworld exercises, one mode of instruction will be to withhold reporting of grammatical errors and respond only to factual and semantic errors, ideally by conventions that occur naturally in conversation. Another mode of instruction is to report all grammatical errors.

We are also aware that when students lose confidence in their ability to reduce errors eventually, the resulting situation is punishing to them, and they give up literally or effectively (Maier, 1949). When they literally give up, they stop practicing and learning. When they effectively give up, they continue to practice, but not to try, and they stop learning. This suggests that the tutor should be designed to reduce the chances of putting the student in situations where errors are overwhelming. Another reason for trying to curb errors is that students appear to learn best when they produce and practice correct behaviors (Anderson, Boyle, Farrell, & Reiser, 1984; Skinner, 1957). That is, learners who manage to avoid errors can more efficiently build desired activities (or mental processes) into their repertoires. The learning principle here is to build for success, an idea put to work in the series of tutors for geometry and for programming designed by Anderson and colleagues (Anderson, Boyle, & Yost, 1985; Anderson, Conrad, & Corbett, 1989), following a model of human cognition known as ACT* (Anderson, 1983). These tutors present material in small increments designed to minimize the errors students can make in progressing to the next step. At the first error, the student is corrected.

Two approaches to containing errors in a tutoring system are:

1. Put the exercises in a fixed sequence, such as from simple to complex, that the user population can ideally traverse without excessive numbers of errors at any one point (as done in programmed learning).
2. Make the tutor adapt to any current state of the individual user's changing level of performance (as done in the ACT* tutors through a process called "model tracing").

The problem with the first approach is that it assumes that the user population is homogeneous and well understood. If this is not the case, then some users will produce excessive errors, while others will find the exercises unchallenging and unrewarding.

The MILT-CSR Microworld and Speech Recognition

This exercise will allow students to use continuous speech to manipulate a graphics microworld. The successful manipulation of the graphics will let users control the microworld environment and should therefore be intrinsically rewarding. In addition,

the microworld can be set in a problem solving domain, and the problem solving itself should provide intrinsic motivation.

Constraining student input. A major problem of continuous speech recognition is that it cannot recognize all speech input, and users may not know what its limits are. One approach to rectifying this situation is to give the students access to some kind of help file which gives them this information. To the extent that the recognizer is limited only to a few utterances, this is a workable solution. However, if there are many utterances that can be recognized and no specific rule to describe what cannot be recognized, then a simple help file will not work well.

The MILT-CSR approach to constraining students' speech input is to present a training version of the microworld in which three alternative recognizable texts are displayed, and the student speaks one of them. The recognized speech triggers animation, and when this is completed, three new, appropriate, recognizable texts are displayed at the bottom of the microworld. The training version of the microworld will be intrinsically interesting though much shorter than the full version. All the major commands, questions, etc. which apply to the continuous recognition version of the microworld will be practiced in the training version. Once the training version is over, text will no longer be displayed and a new microworld exercise will be available. In this way students will be constrained by having learned what the continuous recognizer is capable of in an intrinsically entertaining and useful microworld exercise, and they ought to be significantly less likely to overstep its capabilities.

Interacting with the Microworld. In practice, a student will be able to see a graphic scene and enter a spoken command in the target language, such as "open the briefcase," as shown in Fig 2. Once the spoken command is recognized and turned into ASCII, that ASCII is analyzed in one of two ways, depending upon the version of MILT-CSR in use. In the research version of MILT-CSR a natural language processing (NLP) engine will analyze the entry and converts it to an interlingual representation, using the lexical conceptual structure (LCS) discussed by Dorr et al. (1995). That representation will link to appropriate, animated graphics—for example, the briefcase will open. Because LCS analysis can accommodate some kinds of linguistic equivalence, the tutor will be able to handle different forms of linguistic input to accomplish the same action (e.g., "open the briefcase," "make the briefcase open up," etc.). In the nonresearch version, a string matcher with slots for verbs and objects will link to the animated graphics.

If the student speaks a command that appropriately reflects that student's intention—such as "open the briefcase" to see the briefcase opened—then the student will be able to successfully manipulate the microworld. If student uses the wrong words or constructions to express their intentions, but the entry can still be processed, then the expressed action will take place in the microworld. For example, if the student has not learned the prepositional system in the target language, and says "put the briefcase under the table," then that action will occur, even if the student means to request that the briefcase go on top. If the student speaks a command that cannot be

accomplished in the microworld—if the objects called for are not present, or the properties are not legal—then the student will be given conversationally realistic, discriminative feedback, such as "there is no chair to be moved" (for absent objects) or "the table cannot open" (for illegal or unassigned properties).

The MILT-CSR microworld is partially authorable from a list of objects that can be chosen by authors. The scenario backdrop can be changed as well. Readable objects like books, newspaper, notebooks, envelopes and letters can be rewritten. Objects which produce sound like radios, tape recorders and scenario characters can produce speech via recordable WAV files. Authors easily can create multiple interconnected rooms, and any given room or outside scene can use multiple screens. That is, the student can move his or her animated agent to the left or right, and upon reaching the edge of the screen the next part of the room or outside will be displayed. The number of rooms and the size of rooms or outside scenes is limited only by disk storage. Moreover, authors are able to create either free-play environments or game-like scenarios with specified goals, such as determining the identity of the briefcase owner for the scenario in Figure 2.

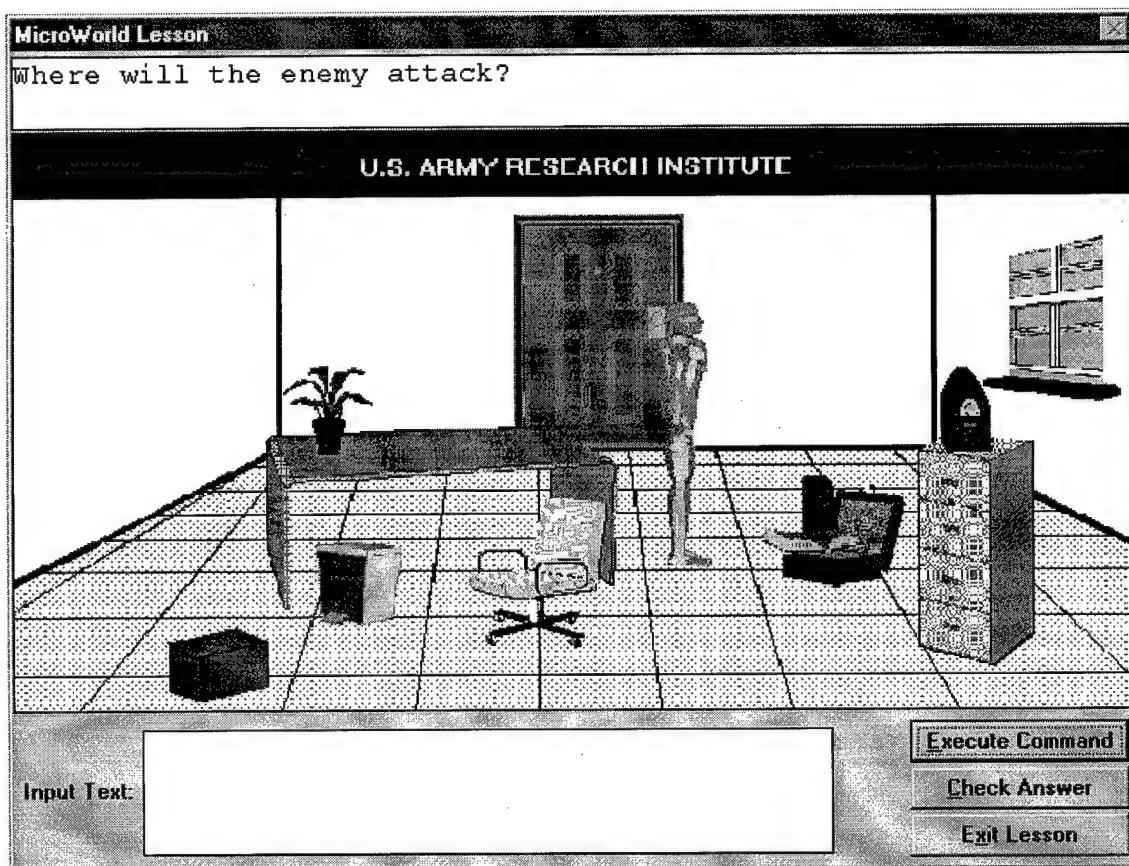


Figure 2. Typical MILT-CSR Microworld exercise screen with open book about to be read

The microworld and syntax errors. The original version of MILT-CSR accepts only text input. This input is sent to a natural language processing (NLP) engine which analyzes it and identifies major classes of grammatical (or syntactic) errors. Being able to tell students what their actual error is rather than just telling them that they have made an error, is a considerable advantage. Installing speech recognition in place of keyboard input either eliminates this possibility or requires a level of recognition accuracy that appears to be beyond the current state of the art unless an errorful language model has been developed. In the medium run, this problem will be solved by the development of such errorful models. It is likely that in the long run, it will be solved by improvements in recognition accuracy. However, in the short run, MILT-CSR deals with this problem through the use of combinations of input. That is, different exercises (including the microworld) will be made available to students some with speech input and others with text input. Since the text input exercises will have a greater diagnostic capability, it will lead to an interesting pedagogical question of how best to combine the two in an overall lesson.

Adapting Exercises to the Learner's Error Performance

To adjust to varying levels of student ability and performance, we designed MILT-CSR to enable faster progress for students who make fewer errors and to enable slower progress, as well as error-specific remediation, for students who make more errors.

Our tutor does not attempt the detailed model tracing employed in the ACT* tutors, but rather keeps a running count of each type of error made and waits for errors to reach some threshold before branching to remedial exercises (following work by Atkinson, 1976; Goldstein, 1979). When errors of a given type reach a prespecified threshold, a remedial set of exercises is automatically triggered. Moreover, this threshold is easily modifiable through the authoring interface.

The adaptive branching in MILT-CSR can employ rules that range from very simple IF and GOTO statements, to computationally complex operations, the building blocks for a formal student model. MILT-CSR supports computationally complex operations by linking performance criteria to Boolean ("and/or") statements. Error counts are compared to settable thresholds, which can be considered as Boolean combinations.

For example, branching to a given kind of remediation can depend on the student's reaching a threshold of subject-verb agreement errors *together with* a threshold of verb tense formation errors. An example of the authoring interfaces that control this adaptive structure appears in Figure 3. In addition to error-based adaptation, MILT-CSR can be modified to allow students to select their next exercise or set,

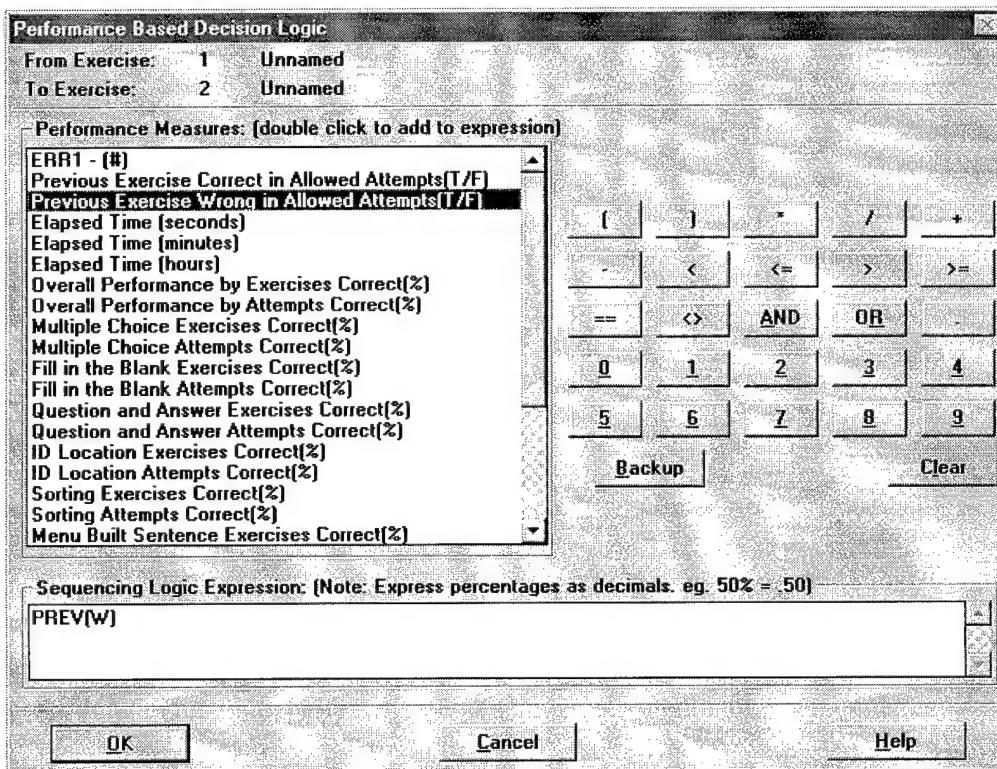


Figure 3. Performance-based sequencing in MILT-CSR showing Boolean approach to controlling adaptivity. (Student's performance on various error types is compared with Boolean combinations of error criteria to determine when to branch from one exercise set to another.)

or it can select the next exercise randomly. The screen that permits this modification is shown in Figure 4.

Parts then the Whole, or Vice Versa

When target skills are complex, teaching component or prerequisite skills to high levels of fluency can lead to faster learning of the complex skill, as found in a variety of tasks (e.g., Stammers, 1980). Indeed, if initial presentations are too complex, learners' attempts to practice may result in failure, anxiety, and eventual unwillingness to continue (Farber & Spence, 1953). In such cases, people learn more effectively if they first practice simpler parts of a task, before putting these parts together into the complete task. This approach is known as *part-task training* and is the basis of many of the training simulators used in the aviation community (Knerr et al., 1985; Wightman & Sistrunk, 1987).

At the same time, some current theories of learning call for starting with the whole task, albeit in simplified form, so that learning is always contextualized. This reasoning underlies constructivist approaches such as *apprenticeship learning* (Collins, Brown, & Newman, 1989). It might also be argued to underlie the immersion method

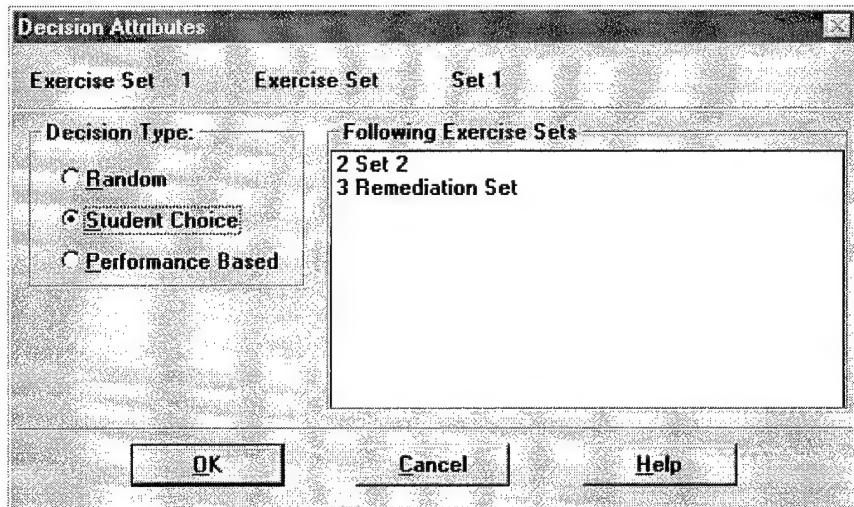


Figure 4. Decision type menu controlling the sequencing of exercises in MILT. *Random* picks the next out of alternative exercises according to a random number generator. *Student choice* allows the student to pick the next exercise. *Performance based* makes use of performance criteria that are set by the author individually, or in combination with a Boolean editor (see Fig. 3).

of language teaching, which stems from the communicative approach (see Oxford, this volume), whereas the grammar drill and practice and audiolingual methods reflect a part-task philosophy

Both theoretical directions suggest that a tutor be designed to provide multiple types of exercises for students. These exercises should provide practice in simplified as well as complex versions of the skill being taught. Following these assumptions, we equipped the MILT-CSR authoring interface with exercise templates that correspond to progressively more cognitively demanding exercise types. Practice in recognizing written or spoken language is possible using multiple choice questions. Production of words and phrases may be practiced using fill-in-the-blank questions. A kind of proto-sentence production is enabled by exercises in assembling sentences from words in a menu (from a set of methods called *guided sentence production* by Kempen, 1992). The free response questions mentioned earlier elicit full sentence production from students. At the most complex end of this continuum are the microworld and dialogue exercises, which permit more extended communicative interaction.

This exercise variety permits both the part-to-whole and the whole-to-part approaches to be defined operationally within the tutor, and then compared and tested. The part-to-whole approach would suggest that the student be offered the opportunity to practice relatively simple, fragmentary language input and understanding, then the more complex full sentence version, and finally, complete dialogue. The whole-to-part approach would suggest that the student (who has already reached an early level of proficiency) start with dialogue and microworld interactions, then go periodically to simpler exercises to practice vocabulary or syntax found to be not yet mastered. Indeed, the

flexibility we have built into this system permits many of the somewhat ethereal notions about language pedagogy (such as "contextualized language teaching") to be brought down to earth and defined within the manipulable parameters of the tutor.

Human Factors in the Authoring Interface

For practical applications, a tutor should be designed either with a very large body of exercises and/or with the ability to accept new exercises. Although we are designing a set of demonstration exercises in Arabic and Spanish, we decided not to install a full curriculum but rather to design an authoring interface so that instructors or researchers could build their own lessons. Unfortunately, most authoring systems are so difficult to learn that they require their own tutor. Therefore, the authoring interface being developed for MILT-CSR is based largely on templates so as to require no programming expertise or other specialized knowledge on the part of lesson authors.

The MILT-CSR authoring system allows foreign language teachers not only to create their own exercises but also to control the sequencing of those exercises. Exercise creation is based on templates for specific exercise types. Authoring conventional exercise types (such as fill in the blank) is handled by a conventional template interface. However, easily creating new microworlds required a unique approach. Figure 5 shows the exercise template for a microworld exercise. The author types desired elements into the various fields, which should require little or no training to understand.

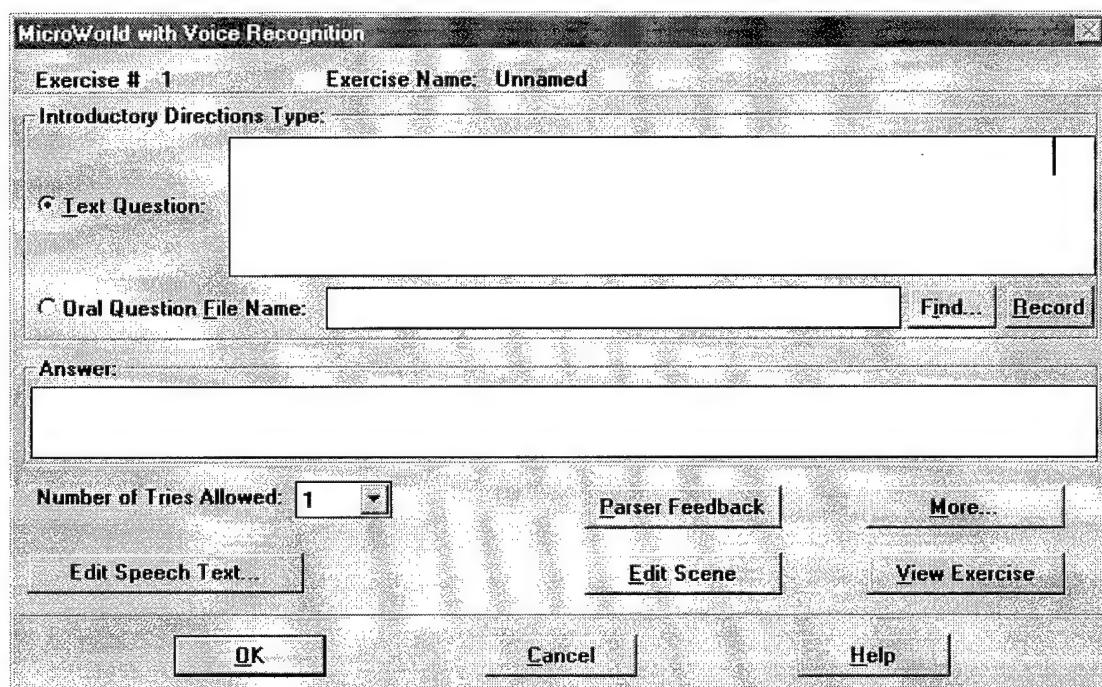


Figure 5. Initial authoring interface for creating a microworld

Figure 6, shows the interface for selecting a background and the objects that will be placed in that background. In this case the background is a graphic entitled room, and the author has the ability to various objects from the Add Object window to place in that Room. Moving up or down allows the author to place objects in front of each other, and set object location allows the author to place the various objects in the exact locations in the room where they will be initially displayed to the student.

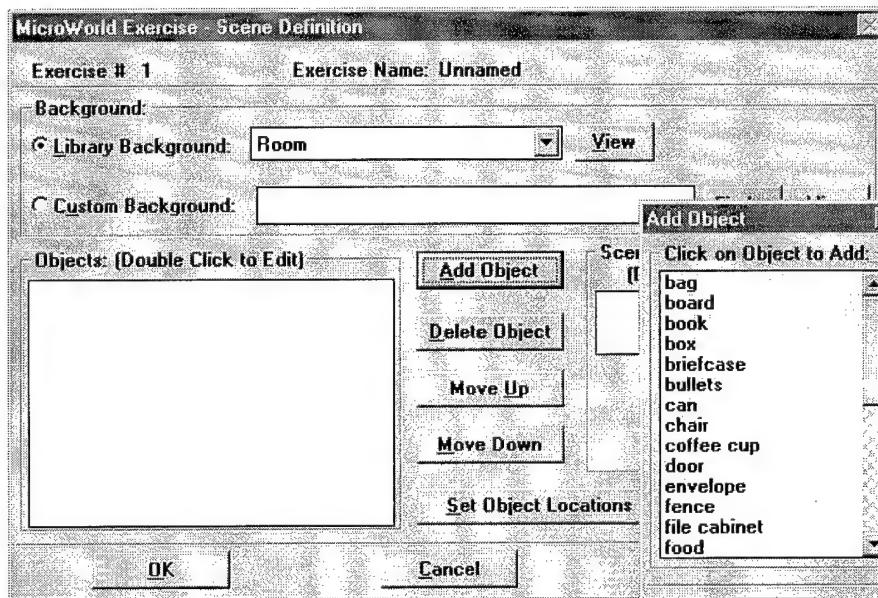


Figure 6. Authoring interface for adding objects to a microworld

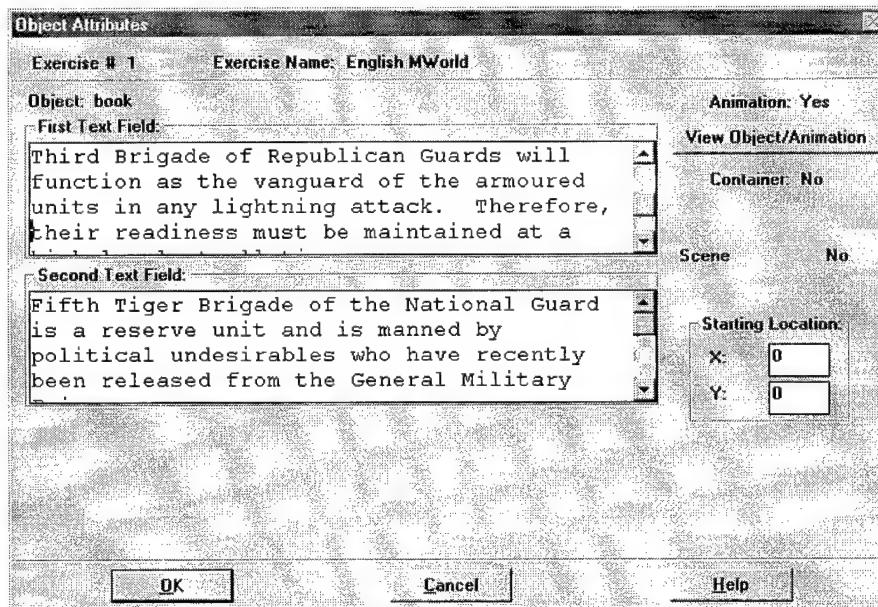


Figure 7. Authoring interface for creating new text for a microworld book

Figure 7 shows the authoring interface for entering new text into the book object. This approach to adding new material applies to many other classes of microworld objects such as books, letters, and newspapers. Other microworld objects can have graphics or sound added to them in the authoring process.

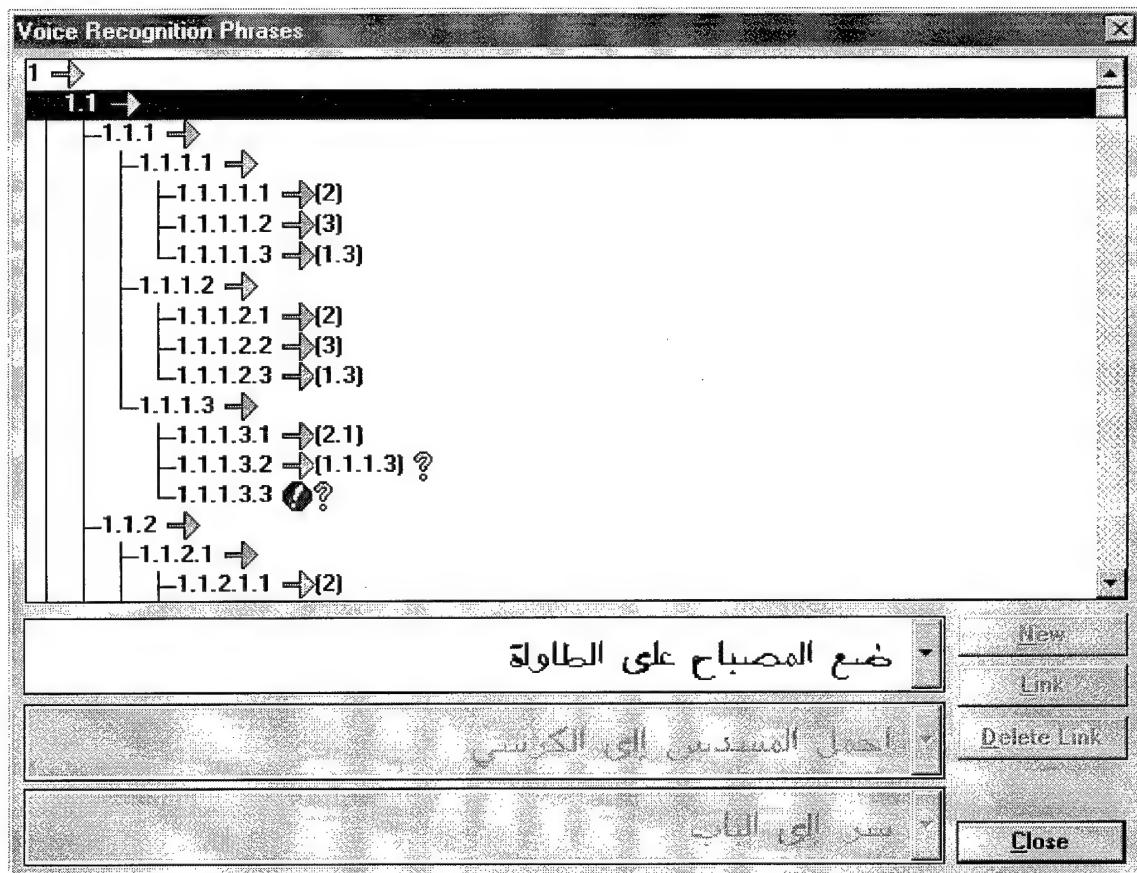


Figure 8. Authoring interface for selecting discrete speech sequence.

Figure 8 shows the interface that allows authors to select the sequence of discrete speech utterances in the MILT microworld. The author can use this screen to select any existing discrete utterance and create sets of three which are displayed at the bottom of student screen (See Figure 9 for the student microworld with speech utterances.). The authors can use branching logic to define the sequences of utterances that will be displayed and activated. That is, when a given utterance is made by the student, the next defined set of three new utterances will be displayed and activated by the speech engine.

Exercise sequences can be fixed and under the control of the author, fixed and under the control of the student, variable based on student performance as interpreted by rules created by the author, or a combination of the above. The basic authoring interface for sequencing uses a flow chart approach, as shown in Figure 10. The decision diamonds in these flow charts indicate points at which the author has specified

branching based on the student's performance. For example, the configuration in Figure 10 means that students who accumulate too many errors and/or too much time upon completion of exercise 3 are branched automatically to exercise 4 instead of exercise 5. When they complete remedial exercise 4 they will be sent to exercise 5. They will have to continue redoing exercise 6, pronunciation, until they complete it to a criterion level at which point that set of exercises will be completed.

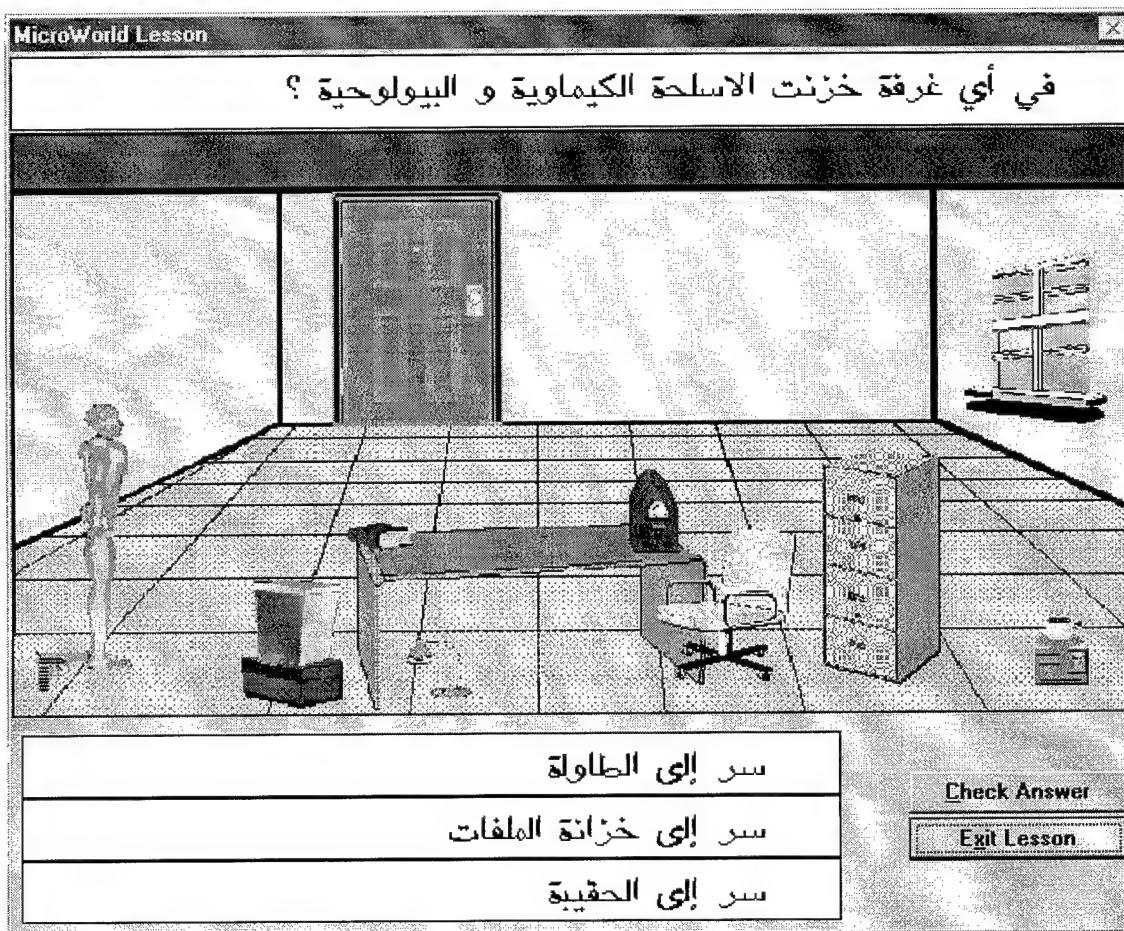


Figure 9. Microworld screen showing Arabic discrete speech utterances from which the student selects.

The threshold that governs the decision diamonds is defined through a separate screen (e.g., Figure 3). Finally, the author can control the timing and wording of exercise feedback through templates.

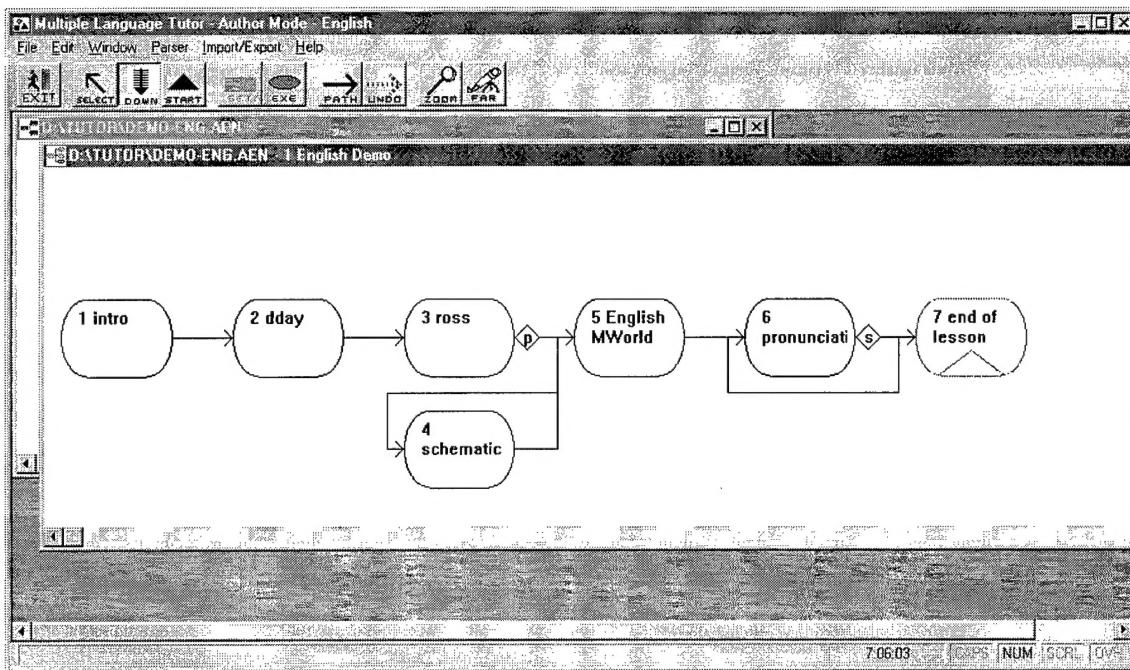


Figure 10. Exercise sequence interface using flow charts to define the progression of exercises and decision diamonds to define branching.

This authoring design is based on an intersection of principles. First, we know that there exists little detail in current theories of language learning and teaching to guide specific features of a tutor like sequencing of exercises and timing of feedback. We therefore built the authoring interface as a research tool to explore these questions. Second, we tried to build in parameters that have been found relevant in research on learning and instruction generally: Sequencing rules and selection and scheduling of feedback have been implicated in numerous studies (see Park & Tennyson, 1983; Schmidt, 1990). Third, we tried to ease the author's burden by using templates where possible.

Final Thoughts

The use of speech recognition in language tutors brings forward a series of issues. Is speech recognition sufficiently accurate to be used? How do you deal with student errors that native speakers don't make? How do you deal with changes in error types, error rates, and pronunciation as students improve? Perhaps most basic of all—how do you use an emerging but imperfect technology like speech recognition to improve language learning? It is our position that the literature regarding the acquisition of cognitive skills can be drawn upon to help answer these basic questions.

There have been few serious attempts to link the design of computer-assisted language instruction to learning research in areas outside of second languages. However, there exists a rich empirical literature regarding how people acquire and

retain cognitive skills. We feel that a principled tutor should draw on this literature, and we have found its principles selectively useful in shaping our ideas for design.

Acknowledgments

ARI's prime contractor for the MILT-CSR project is Micro Analysis & Design of Boulder, CO. The natural language processing components are developed by the University of Maryland, Institute for Advanced Computer Studies. At present, it makes use of a Dragon discrete speech recognition engine. Its Arabic language model was developed by Interactive Drama. The views, opinions, and conclusions contained in this paper are those of the authors and should not be construed as an official Department of the Army position, policy, or decision.

References

Anderson, J.R.; Boyle, C.F.; Corbett, A.A., & Lewis, M.W. (1990). Cognitive Modeling and Intelligent Tutoring. *Artificial Intelligence* 42, 7-49.

Anderson, J.R., Boyle, C.F., & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), *Proceedings of the Ninth International Conference on Artificial Intelligence* (pp. 1-7). Los Altos, CA: Morgan Kaufman.

Anderson, J.R., Conrad, F., & Corbett, A. (1989). Skill acquisition and the LISP tutor. *Cognitive Science*, 8, 87-129.

Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-410.

Anderson, R.C., Kulhavy, R., & Andre, T. (1971). Feedback procedures in programmed instruction. *Journal of educational research*, 62, 148-156.

Atkinson, R.C. (1976). Adaptive instructional systems: Some attempts to optimize the learning process. In Klahr, D. (Ed.), *Cognition and instruction*. Hillsdale, NJ: Lawrence Erlbaum.

Berkowitz, M., & Simutis, Z. (1983). The development of a hand-held computerized tutor for military vocabulary. *Journal of computer-based instruction*, 5, 41-44.

Berlyne, D. (1968). Curiosity and exploration. *Science*, 153, 25-33.

Bransford, J. D., & Franks, J. J. (1971). The abstraction of linguistic ideas. *Cognitive Psychology*, 2,

Chomsky, N. (1981). *Lectures on government and binding*. Retention of military tasks: A review. *Human Factors*, 25, 199-213.

Holland, V.M. (forthcoming). Lessons learned in developing intelligent CALL: Designing across disciplines. To appear in *Computer Assisted Language Learning*.

Hull, C.L. (1951). *Essentials of behavior*. New Haven: Yale University Press.

Kaplan, J.D. (Ed.) (1989). *MANPRINT methods monograph: Aiding the development of training constraints* (Technical Report NO 890621). Alexandria, VA: U.S. Army Research Institute.

Kempen, J. (1992). Language technology and language instruction: Computational diagnosis of word-level errors. In M. L. Swartz & M. Yazdani (Eds), *Intelligent tutoring systems for foreign language learning: The bridge to international communication*. New York: Spring-Verlag.

Knerr, C.M., Morrison, J.E., Mumaw, R.J., Stein, D.J., Sticha, P.J., Hoffman, R.G., Buede, D.M., & Holding, D.H. (1985). *Simulator-based research in part-task training* (Report FR-PRD-85-11). Alexandria, VA: HumRRo.

Krueger, W.C.F. (1929). The effects of overlearning on retention. *Journal of General Psychology*, 12, 71-78.

Kulhavy, R.W. (1977). Feedback in written instruction. *Review of Educational Research*, 47(1), 211-232.

Kulik, J.A., & Kulik, C.C. (1988). Timing of Feedback and Verbal Learning. *Review of Educational Research*, 58(1), 79-97.

Kurland, L., & Tenney, Y. (1989). Issues in developing an intelligent tutor for a real-world domain: Training in radar mechanics. In Psotka, J., Massey, D., & Mutter, S. (Eds.), *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Lawrence Erlbaum.

Landauer, T.K., & Bjork, R.A. (1978). Optimum rehearsal patterns and name learning. In M. Gruneberg, P. Morris, & R. Sykes (Eds.), *Practical aspects of memory*. London: Academic Press.

Laughery, K.R., Dahl, S., Kaplan, J.D., Archer, R., & Fontenelle, G. (1988). *Proceedings of the Human Factors Society 32nd Meeting*, 2. Santa Monica, CA: Human Factors Society.

Lawton, G., & Andreson, J.T. (forthcoming). *Toward a behavior analytic assault on foreign language instruction*. Manuscript in preparation. Alexandria, VA: U.S. Army Research Institute, and Durham, NC: Duke University.

Levine, M. (1975). *A cognitive theory of learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lightbown, P., & Spada, M. (1990). Focus on form and corrective feedback in communicative language teaching: effects on second language learning. *Studies in Second Language Acquisition*, 12, 4.

Maier, N.R.F. (1949). *Frustration: The study of behavior without a goal*. New York: McGraw-Hill.

Malone, T.W. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science*, 4, 333-369.

McClelland, D. C. (1961). *The achieving society*. New York: Van Nostrand.

Melton, A. W. (1967). Repetition and retrieval from memory. *Science*, 158, 532.

Miller, N. (1963). Some reflections on the law of effect produce a new alternative to drive reduction. In M. R. Jones (Ed.), *Nebraska Symposium on Motivation*. Lincoln: University of Nebraska Press.

Mowrer, O. (1960). *Learning theory and the symbolic processes*. New York: Wiley.

Newell, A. (1990). *A unified theory of cognition*. Cambridge, MA.: Harvard University Press.

Oxford, R., Harman, J., & Holland, V.M. (1987). Advances in the development of hand-held, computerized game-based, training devices. In Crookall, D., & Greenblatt, C. (Eds.), *Simulation-gaming in the late 1980s*. New York: Pergamon Press.

Park, O.K., & Tennyson, R.D. (1983). Computer-based instructional systems for adaptive education: A review. *Contemporary education review*, 2, 121-134.

Psotka, J., Massey, D., & Mutter, S. (Eds.) (1988). *Intelligent tutoring systems: Lessons learned*. Hillsdale, NJ: Lawrence Erlbaum.

Schendel, J.D., & Hagman, J.D. (1982). On sustaining procedural skills over a prolonged retention interval. *Journal of Applied Psychology*, 67, 605-610.

Schmidt, R.A. (1990). *Feedback for skill acquisition: Preliminaries to a theory of feedback* (Final Report, U.S. Army Research Institute Contract MDA903-85-K-0225). University of California at Los Angeles: Dept. of Psychology Motor Control Laboratory.

Schmidt, R.A., & Bjork, R. (1992). New conceptualizations of practice. *Psychological Science*, 3(4), 207-217.

Skinner, B.F. (1938). *The Behavior of Organisms*. New York: Appleton-Century.

Skinner, B. F. (1957). *Verbal behavior*. New York: Appleton-Century-Crofts.

Stammers, R.B. (1980). Part and whole practice for a tracking task: Effects of task variables and amount of practice. *Perceptual and Motor Skills*, 50, 203-210.

Wightman, D.C., & Sistrunk, F. (1987). Part-task training strategies in simulated carrier landing final-approach training. *Human Factors*, 29 (3), 245-254.

Wisher, R.A., Holland, V.M., & Chatelier, P. (May, 1987). *Portable training and job aiding devices*. Paper presented at the 1987 NATO Conference on Military Personnel and Training, Luxembourg (published in the proceedings).

Wisher, R.A., Sabol, M., & Kern, R. (forthcoming). Modeling acquisition of an advanced skill: The case of Morse Code copying. To appear in *Instructional Science*.

Witkins, H.A., Moore, C.A., Goodenough, D., & Cox, P. (1977). Field-independent and field-dependent cognitive styles and their educational implications. *Review of educational research*, 47, 1-64.

Yates, L., & Macpherson, D. (1985). Analytic prediction of training device effectiveness. In *Proceedings of the 27th Annual Conference of the Military Testing Association* (pp. 515-520). San Diego, CA: U.S. Navy Personnel Research and Development Center.